Fig. 1

200

DISPLAY
204

PROCESSOR ～206

VERIFICATION
MODULE ～102

RAM ～208

VOB ～112
CODE_FILE ～101
MODEL_FILE ～103
OUTPUT ～108

～210

Fig. 2

300

in1 (f,0,L) [1] ~302

in2 (f,0,L) [2] ~304

in3 (f,0,L) [3] ~306

308
| + |
| − |
| + |
sum

out1 (?) [1] 310

318
1/z
simpleDelay
{IC=5}

pi
constant
c=double(3.14159)
314

312
X
product

out2 (?) [2] 316
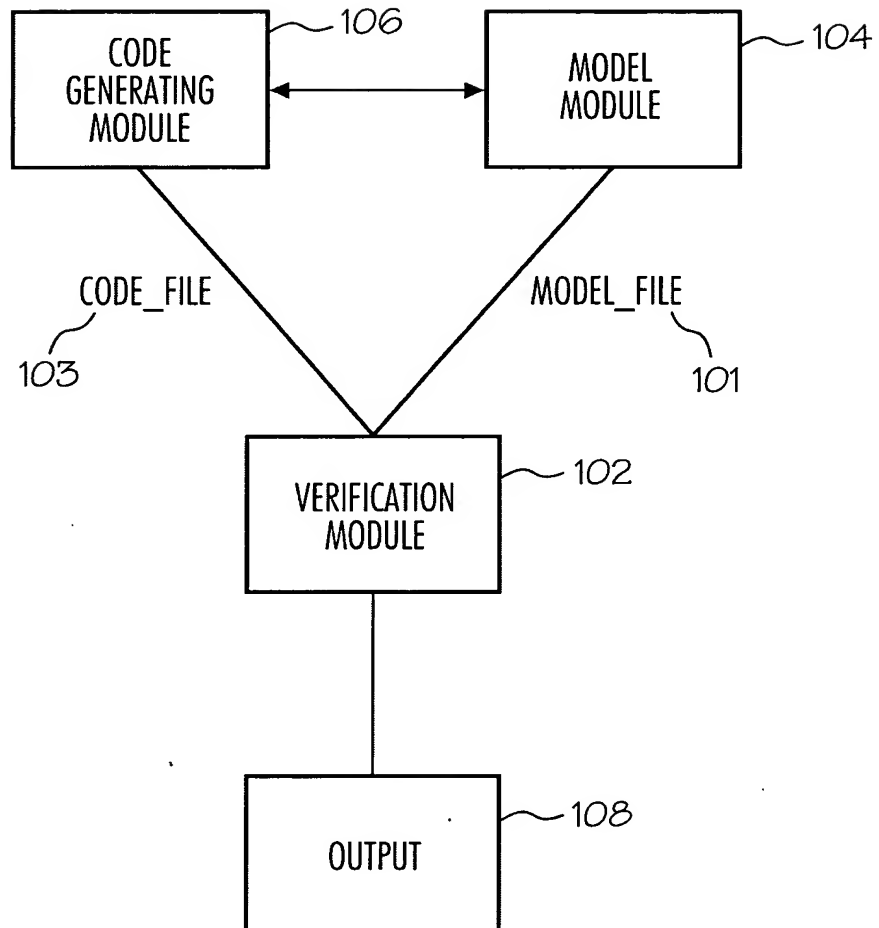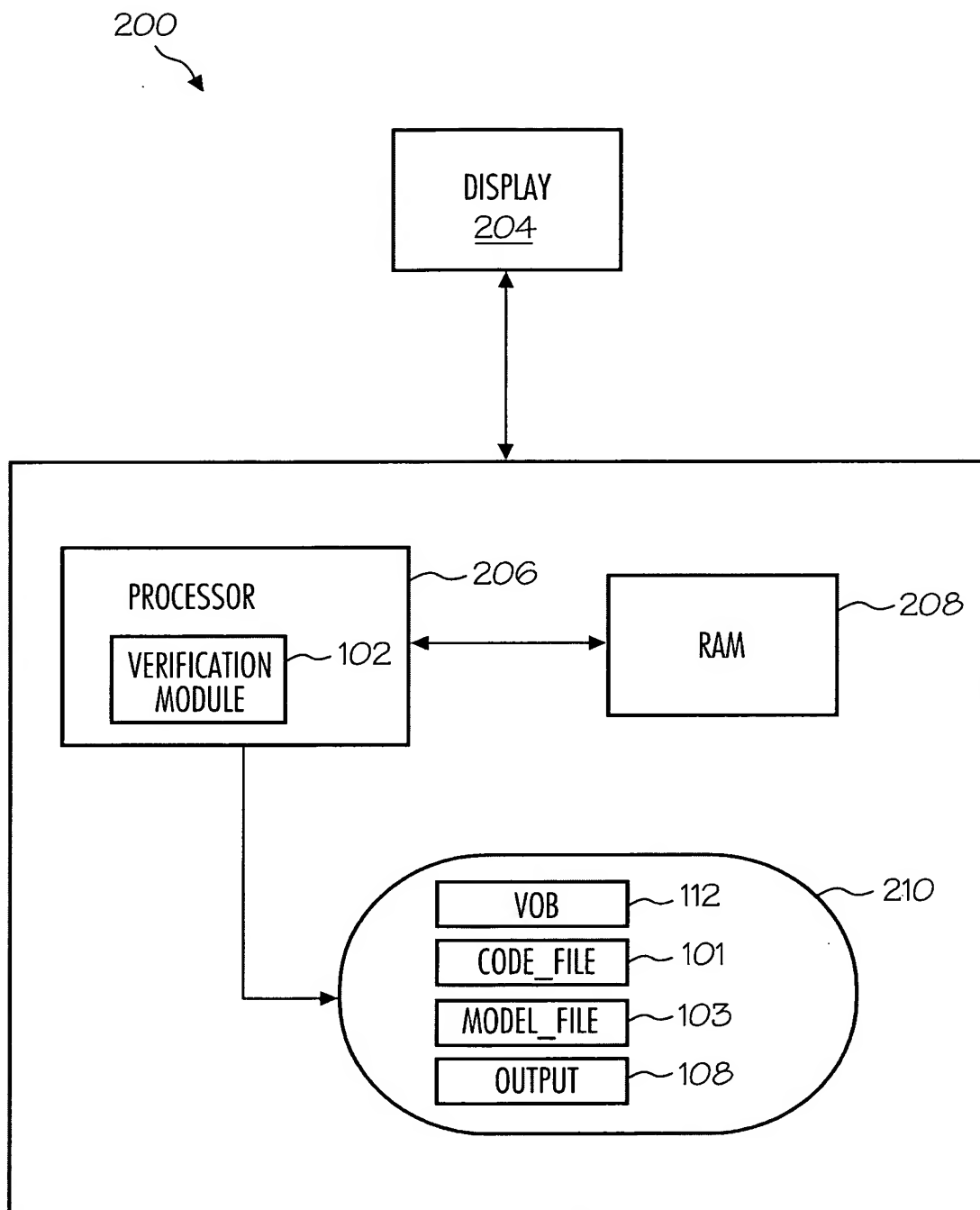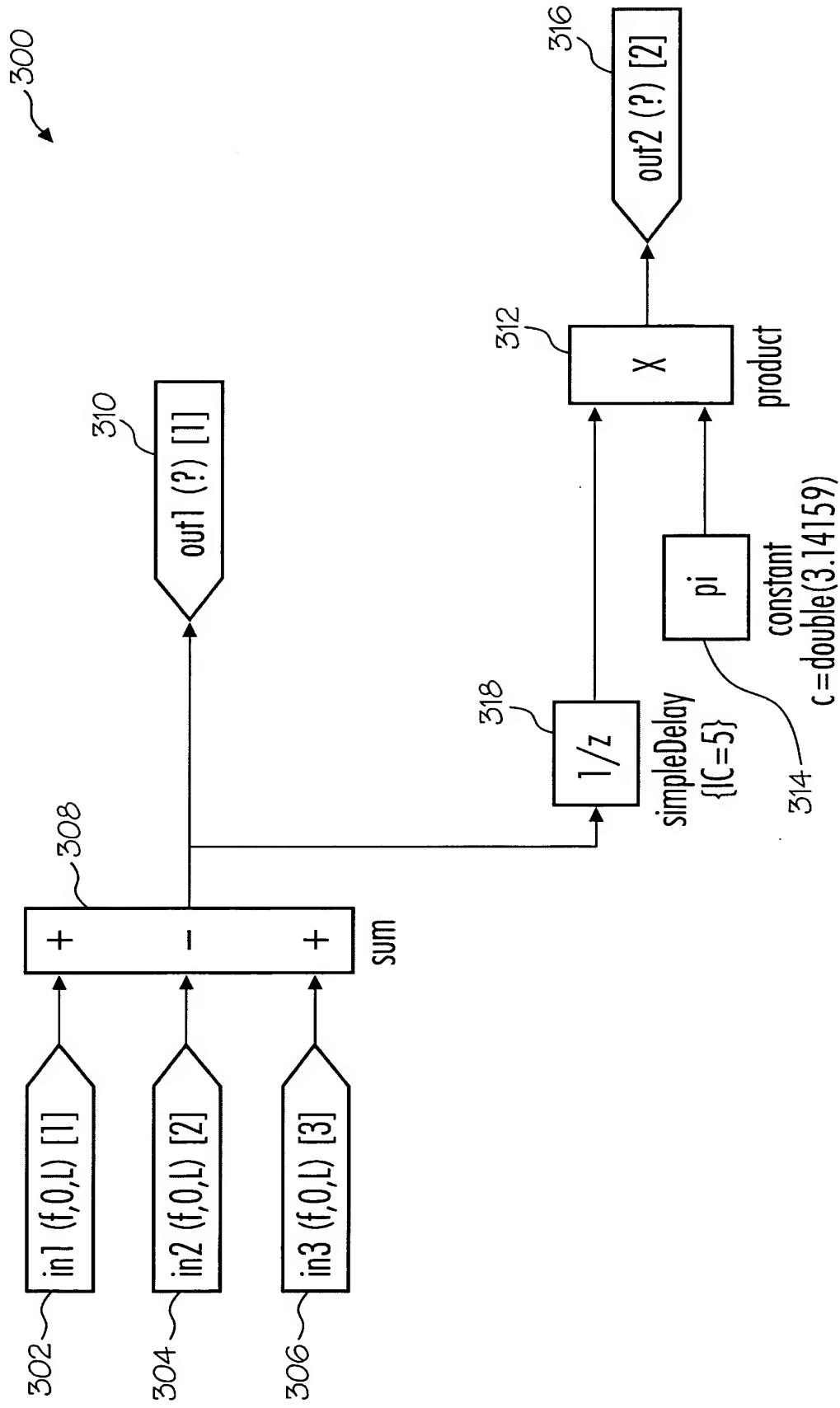
Fig. 3a

320

Ccrdemo.c:

322

```
/*
 * COMPANY X PROPRIETARY AND CONFIDENTIAL
 * Copyright (c) 1996-2004 Company X International Inc.
 * Unpublished Work.  All rights reserved.
 */

const char proprietaryNotice[] = "* COMPANY X PROPRIETARY AND CONFIDENTIAL  * Copyright
(c) 1996-2004 Company X International Inc. * Unpublished Work.  All rights reserved.";

/*
 * Real-Time Workshop code generation for Simulink model "ccrdemo.mdl".
 *
 * Model Version              : 1.4
 * Real-Time Workshop file version    : 4.1.1 $Date: 2001/08/24 12:43:38 $
 * Real-Time Workshop file generated on : Mon Jan 19 12:12:31 2004
 * TLC version                : 4.1 (Oct  5 2001)
 * C source code generated on        : Mon Jan 19 12:12:31 2004
 */

#include "ccrdemo_private.h"
#include "ccrdemo.h"
```

330

```
/* Block parameters (auto storage) */
const Parameters_ccrdemo ccrdemo_P = {
  5.0 ,                 /*SimpleDelay_1_X0 : '<S3>/SimpleDelay_1' */
  3.14159               /*Constant_1_Value : '<S1>/Constant_1' */
};
```

340

```
/* Block signals (auto storage) */
BlockIO_ccrdemo ccrdemo_B;

/* Block states (auto storage) */
D_Work_ccrdemo ccrdemo_DWork;

/* External inputs (root inport signals with auto storage) */
ExternalInputs_ccrdemo ccrdemo_U;

/* External output (root outports fed by signals with auto storage) */
ExternalOutputs_ccrdemo ccrdemo_Y;

/* Real-time object */
ccrdemo_RT_OBJECT ccrdemo_rtO;
```

332

Fig. 3b

```
/* Model step function */
void ccrdemo_step(void)
{
  /* Sum Block: '<S4>/Sum_1' */                          342
  ccrdemo_B.Sum_1 = ccrdemo_U.in1 - ccrdemo_U.in2 + ccrdemo_U.in3;

  /* Outport Block: '<Root>/out1' */
  ccrdemo_Y.out1 = ccrdemo_B.Sum_1;

  /* UnitDelay Block: '<S3>/SimpleDelay_1' */
  ccrdemo_B.SimpleDelay_1 = ccrdemo_DWork.SimpleDelay_1_DSTATE;

  /* Product Block: '<S2>/Product_1' */
  ccrdemo_B.Product_1 = ccrdemo_B.SimpleDelay_1 * ccrdemo_P.Constant_1_Value;

  /* Outport Block: '<Root>/out2' */
  ccrdemo_Y.out2 = ccrdemo_B.Product_1;
```
```
  /* UnitDelay Block: <S3>/SimpleDelay_1 */
  ccrdemo_DWork.SimpleDelay_1_DSTATE = ccrdemo_B.Sum_1;
}
```
```
/* Model initialize function */
void ccrdemo_initialize(boolean_T firstTime)
{
 if (firstTime) {
   /* registration code */
   ccrdemo_rt0.errorStatus = (void *)0;

   /* UnitDelay Block: <S3>/SimpleDelay_1 */
   ccrdemo_B.SimpleDelay_1 = ccrdemo_P.SimpleDelay_1_X0;
 }

 /* UnitDelay Block: <S3>/SimpleDelay_1 */
 ccrdemo_DWork.SimpleDelay_1_DSTATE = ccrdemo_P.SimpleDelay_1_X0;
}
```
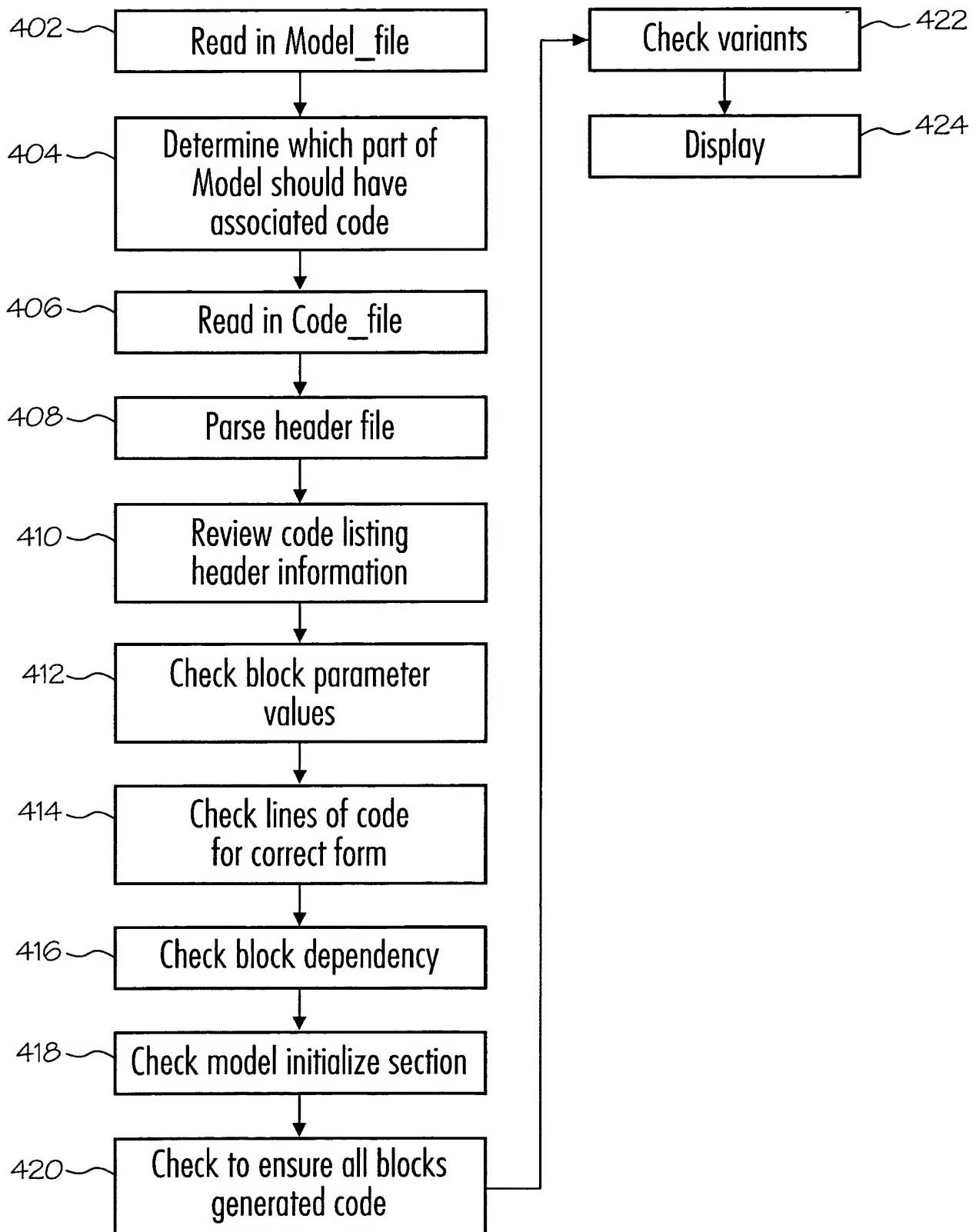
334

336

338

Fig. 3b (continued)

402 — Read in Model_file

404 — Determine which part of Model should have associated code

406 — Read in Code_file

408 — Parse header file

410 — Review code listing header information

412 — Check block parameter values

414 — Check lines of code for correct form

416 — Check block dependency

418 — Check model initialize section

420 — Check to ensure all blocks generated code

422 — Check variants

424 — Display

Fig. 4